In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown in accordance with the mandatory amendment format.

1-2. (Cancelled)

3. (Currently Amended) A method comprising:

reading data from a first dirty cache line in a cache memory;

determining if the data is corrupt;

marking the first dirty cache line invalid if the data is corrupt;

determining if a duplicate cache line exists;

determining if the data within the duplicate cache line is corrupt if the duplicate cache line exists;

writing the data within the duplicate cache line to a first location in memory if the duplicate cache line is not corrupt; and

marking the first dirty cache line available.

4. (Currently Amended) The method of claim 3, further comprising:

if the data from the first dirty cache line is not corrupt, then:

writing the data from the first dirty cache line to the first memory location ~~if the data is not corrupt~~;

marking the first dirty cache line available;

determining at least one duplicate dirty cache line ~~for~~ of the first dirty cache line; and

marking each duplicate dirty cache line as an available cache line.

5. (Previously Presented) The method of claim 3, further comprising marking each duplicate dirty cache line invalid if the data within the duplicate dirty cache line is corrupt.

6. (Previously Presented) The method of claim 3, further comprising determining that no duplicate cache lines exist.

7. (Previously Presented) The method of claim 6, further comprising terminating the method if a duplicate cache line is not found.

8. (Currently Amended) A method comprising:

reading data from a first cache line in a cache memory;

determining if the first cache line is a ~~clean~~ dirty line;

determining if the data in the first cache line is corrupt;

marking the first cache line invalid if the data is dirty and if the data is corrupt;

determining if a duplicate cache line exists;

determining if the data within the duplicate cache line is corrupt if the duplicate cache line exists;

writing the data within the duplicate cache line to a first location in memory if the duplicate cache line is not corrupt; and

marking the first ~~dirty~~ cache line available.

9. (Currently Amended) The method of claim 8, further comprising:

~~ii. no more duplicate cache lines exist.~~

marking the cache line invalid if the data is corrupt and the first cache line is not clean;

determining if a duplicate cache line exists;

determining if the data is corrupt if a duplicate cache line exists;

writing the data to a first memory location if the data is not corrupt; and

marking the first cache line available. ~~; and~~

10. (Previously Presented) The method of claim 9, further comprising terminating the method if a duplicate cache line is not found.

11. (Previously Presented) The method of claim 10, further comprising reading from a second cache line if the data is corrupt, and the first cache line is clean.

12. (Currently Amended) The method of claim 3, wherein ~~each of~~ the duplicate ~~dirty~~ cache line[[s]] is marked invalid.

13. (Previously Presented) The method of claim 9, further comprising determining that no duplicate cache lines exist.

14-19. (Cancelled)

20. (Currently Amended) An apparatus comprising:

a cache memory; and

a cache controller, coupled to the cache memory, to receive a request to write data to a location within the cache memory, read data from a first dirty cache line in the cache memory in response to receiving the request, mark the first dirty cache line invalid if the data is corrupt, determine if a duplicate cache line exists, write the data to a first location in a main memory device if the duplicate cache line is not corrupt, and mark the first dirty cache line available.

21. (Currently Amended) The apparatus of claim 20, wherein the cache controller invokes a

replacement policy to free up one or more cache lines ~~in the associated set~~ if there are no cache lines available.

22. (Currently Amended) The apparatus of claim 20, wherein the cache controller further writes the data to the first memory location if the data from the first dirty cache line is not corrupt, marks the first dirty cache line available, determines at least one duplicate dirty cache line for the first dirty cache line, and marks each duplicate dirty cache line as an available cache line.

23-24. (Cancelled)